

REMARKS

[0003] Applicant respectfully requests entry of the following remarks and reconsideration of the subject application. Applicant respectfully requests entry of the amendments herein. The remarks and amendments should be entered under 37 C.F.R. §1.116 as they place the application in better form for appeal, or for resolution on the merits.

[0004] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-9 and 11-54 are presently pending. Claims amended herein are: 1, 2, 4, 8, 13, 18, 25, 31-36, 40, 46, 48, 50, and 51.

Statement of Substance of Interview

[0005] The Examiner graciously talked with me—the undersigned representative for the Applicant—on May 21, 2008. Applicant greatly appreciates the Examiner’s willingness to talk. Such willingness is invaluable to both of us in our common goal of an expedited prosecution of this patent application.

[0006] During the interview, I discussed examples of how the claims differ from the cited references. The Examiner was receptive to the proposals, and I understood the Examiner to tentatively concur that the proposed claim amendments appeared to distinguish over the cited references of record. For example, the Examiner indicated that clarification regarding “generic class” under the context of object-oriented programming language specification, when considered in view of the claim amendment, might distinguish over the cited references. The Examiner requested that a formal response be presented in writing for further consideration.

[0007] Applicant herein amends the claims in the manner discussed during the interview. Accordingly, Applicant submits that the pending claims are allowable over the cited references of record for at least the reasons discussed during the interview.

Formal Request for an Interview

[0008] If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0009] Please contact me to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for me, I welcome your call as well. My contact information may be found on the last page of this response.

Claim Amendments

[0010] Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1, 2, 4, 8, 13, 18, 25, 31-36, 40, 46, 48, 50, and 51 herein. These amendments are fully supported by the Application and therefore do not constitute new matter. Accordingly, entry to the file is respectfully requested.

[0011] Applicant amends the claims to clarify claimed features. Such amendments are made to expedite prosecution and more quickly identify allowable subject matter. Such amendments are merely intended to clarify the claimed features, and should not be construed as further limiting the claimed invention in response to the cited references.

Formal Matters

[0012] This section addresses any formal matters (e.g., objections) raised by the Examiner

Provisional Double-Patenting Rejections

[0013] Based upon co-pending application 10/657,468, the Examiner rejects claims 1, 13 and 18 on the grounds of non-statutory obviousness-type double-patenting. Applicant respectfully requests the objection be held in abeyance until either of the applications matures to a patent.

Substantive Matters

Claim Rejections under § 103

[0014] Claims 1-9, 11-54 are rejected under 35 U.S.C. §103(a) for being unpatentable over a variety of combinations of cited references including U.S. Patent Application Publication 2005/0060695 to Hostetter et al. (“Hostetter”), an article “High-performance parallel programming in Java exploiting native libraries” by Getov et al. (“Getov”), an article “Product Snapshot: J#, J# provides Java developers a key for entering the .Net platform” by Johnthan Lurie (“Lurie”), and “Diagnosing Java code: Java generics without the pain” by Eric Allen (“Allen”).

[0015] In light of the amendments presented herein and the decisions/agreements reached during the above-discussed Examiner interview, Applicant submits that these rejections are moot. Accordingly, Applicant asks the Examiner to withdraw these rejections.

[0016] Independent claim 1, as amended, recites (Emphasis added):

1. A method of generating common intermediate language code for use in a framework, the method comprising:

receiving a portion of JAVATM language source code *referencing, through a generic class syntax, one or more generic classes unspecified in a formal JAVATM language specification*, wherein:

each of the one or more generic classes refers to a first class configured to operate uniformly on values of different types associated with the first class and defined by a plurality of second classes;

the generic class syntax is not supported by the formal JAVATM language specification and identifies one of the plurality of second classes by surrounding the one of the plurality of second classes with angular brackets following the first class; and

...

generating, through a first compiler different from a formal compiler complying with the formal JAVATM language specification, language-neutral intermediate language code representing the portion of JAVATM language source code referencing the one or more generic classes.

[0017] It is known in the art that “Generic Classes” refer to a particular term in object-oriented programming language specification that, when implemented by one or more object-oriented programming languages, encapsulate operations not specific to a particular data type (i.e., classes, interfaces and methods that operate uniformly on values of different types such that adding and/or removing items from the generic classes are performed in almost the same way regardless of the type of data being stored). In order for a programming language to recognize and support generic classes, certain syntax is specified. For example, formal specification for C++ and other languages set forth generic class syntaxes that specify how generic classes are defined and declared.

[0018] However, at the time the instant invention was conceived, a formal specification for JAVATM language (known as JAVATM Development Kit (JDKTM) 1.1.4) did not specify generic classes (although the latest JDK implements similar feature called “generics”). For example, it’s disclosed in the Application that “[f]ormal specifications for some languages, such as JAVATM language, do not specify generic classes. Thus, generic classes that may be provided in frameworks, or other software packages, are not

readily accessible by developers of JAVATM language source code. For example, currently, JAVATM language source code cannot use a generic class that may be provided by the .NETTM Framework.” (Specification at p.2, lines 11-16). “[A]s used herein, the term “JAVATM language” refers to any source code language that is based on a formal JAVATM language specification, such as, but not limited to, the JAVATM Development Kit (JDKTM) 1.1.4. Although formal JAVATM specifications do not specify generic classes...” (Specification at p.14, line 25 – p.15, line 4).

[0019] Claim 1 was rejected on the combination of Hostetter in view of Getov. Applicant respectfully submits that these rejections are rendered moot in view of the claim amendments. In particular, none of the cited references teach implementing a “generic class” in JAVATM language and specifying syntax for generic classes in JAVATM language.

[0020] Hostetter is directed to a compiling method in JAVATM language, wherein template-generated classes in program code are compiled through a process of lazy compilation, thus improving the compilation time during execution. According to Hostetter, with the template-generated classes, the unnecessary object code results are never invoked during the execution. Hostetter further introduced lazy compilation, which delays compilation of a referenced method until the class method is invoked by the execution of a method call instruction.

[0021] However, all the teachings in Hostetter for compiling source code are based on JAVATM language specification effective at a time that predates the filing date of the instant Application. As the Applicant elaborates above, during the time when the instant invention was conceived, JAVATM language specification did not specify generic classes, nor did it provide specific syntax for generic classes in JAVATM language. Therefore, since

Hostetter provides a solution implemented in JAVATM language that was fully consistent with the JAVATM language specification in effect at that time (i.e., JDK 1.1.4 or earlier) without an intent to incorporate generic classes into the formal JAVATM language specification, Hostetter does not teach, in JAVA language source code, **a code portion referencing one or more generic classes through a generic class syntax** that is unspecified in the formal JAVATM language specification in effect when the instant invention was conceived. (Emphasis added).

[0022] Getov does not cure the deficiency of Hostetter. Getov describes how new Java programs can capitalize on high-performance libraries implemented by other languages (i.e., C or Fortran). By providing a prototype tool for automating the creation of portable interfaces from JAVATM language to native libraries created for MPI, BLAS, ScaLAPACK, etc, Getov concludes that an efficient numerical programming in JAVATM language utilizing these native code libraries is feasible.

[0023] Getov is completely silent in implementing generic classes in JAVATM language. Getov at best teaches creating JAVA binding for MPI, BLAS, ScaLAPACK, etc. to be available for use by the JAVATM language under certain conditions.

[0024] Accordingly, the features including implementing a generic class and providing generic class syntax in JAVATM language are absent in the combination of Hostetter and Getov. Therefore, amended claim 1 is respectfully asserted patentable over the cited references.

[0025] Similarly, since independent claims 13, 18, 25, 31, 36, 40, and 51 incorporate at least the recited feature above, it's respectfully asserted that these claims are also

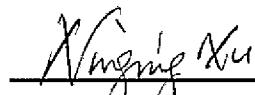
patentable over the cited references for at least the reasons presented above with reference to claim 1.

Conclusion

[0026] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action.** Please call/email me or my assistant at your convenience.

Respectfully Submitted,

Lee & Hayes, PLLC



Ningning Xu (ningning@leehayes.com; x226)

Registration No. L0293

Dated: 2008-05-28

Bea Koempel-Thomas (bea@leehayes.com; x259)

Registration No. 58,213

Customer No. **22801**

Telephone: (509) 324-9256

Facsimile: (509) 323-8979

www.leehayes.com